实验三

knn算法（已实现）

算法思路

将测试和训练的mnist图片从28X28矩阵转为1X784的向量。 每一个测试的数据算出它与所有训练数据的距离，选取距离最小的k个训练数据，从这个k个训练数据中取最多出现的标签为该测试数据的标签。

此处训练数据为20000，测试数据为100，k选取的10.

代码

```
#coding:utf-8

import numpy as np
from datetime import datetime
from tensorflow.examples.tutorials.mnist import input_data
mnist = input_data.read_data_sets('MNIST_data', one_hot=True)


def kNN(Input, dataSet, labels, k):
    num = dataSet.shape[0]
    init_shape = Input.shape[0]
    Input = Input.reshape(1, init_shape) #将矩阵转换为1X784的向量
    diff = np.tile(Input, (num, 1)) - dataSet #将测试数据复制到与训练数据个数相同，再减去训练数据得到距离
    squaredDiff = diff ** 2
    squaredDist = np.sum(squaredDiff, axis = 1)
    distance = squaredDist ** 0.5 #二范式距离
    sortedDistIndices = np.argsort(distance)

    classCount = np.zeros(10)
    for i in xrange(k): #前k个最小距离的训练数据中 分别记录它们的标签
        voteLabel = labels[sortedDistIndices[i]]
        voteLabel=np.argmax(voteLabel,0)
        classCount[voteLabel] =classCount[voteLabel] + 1

    maxIndex=np.argmax(classCount,0)#选取最多的标签为测试数据的类
    print maxIndex
    return maxIndex

def testHandWritingClass():
    batch = mnist.train.next_batch(20000)
    train_x = batch[0]
    train_y = batch[1]

    batch1=mnist.train.next_batch(100)
    test_x = batch1[0]
    test_y = batch1[1]

    a = datetime.now()
    numTestSamples = test_x.shape[0]
    matchCount = 0
    test_num = numTestSamples

    for i in xrange(test_num):
        predict = kNN(test_x[i], train_x, train_y, 10)
        if predict == np.argmax(test_y[i]):
            matchCount += 1

    accuracy = float(matchCount) / test_num
    b = datetime.now()
    print "一共运行了%d秒"%((b-a).seconds)
```

```
    print 'The classify accuracy is: %.2f%%' % (accuracy * 100)


testHandWritingClass()
```